

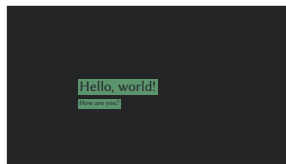
ppt-slides: L^AT_EX Package for Slide Decks à la Power Point*

Yegor Bugayenko
yegor256@gmail.com

2022-12-04, 0.2.1

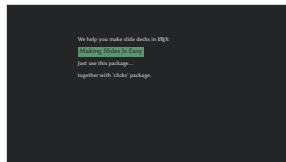
1 Introduction

This package helps you render slide decks in PowerPoint™ style:



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \begin{pptMiddle}
5 \pptTitle{Hello, world!}{How are you?}
6 \end{pptMiddle}
7 \end{document}
```

It is recommended to use this package together with [clicks](#):



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \usepackage[static]{clicks}
4 \begin{document}
5 We help you make slide decks in \LaTeX:
6 \plick{\pptBanner{Making Slides Is Easy}}
7 \plick{Just use this package...}
8 \plush{together with 'clicks' package.}
9 They will look similar to what PowerPoint can
10 make, but with the precision of \LaTeX. We
11 recommend using us together with ‘clicks.’
12 \end{document}
```

2 Layout

```
\pptToc First, split your story into chapters:
\pptChapter
\documentclass{article}
\usepackage{clicks}
\usepackage[template,scheme=light]{ppt-slides}
\begin{document}
```

*The sources are in GitHub at [yegor256/ppt-slides](https://github.com/yegor256/ppt-slides)

```

\pptToc
\plush{\pptChapter{About Me}}
...
\plush{\pptChapter[Idea]{My Idea Is Novel}}
...
\plush{\pptChapter[FAQ]{Discussion \& Questions}}
...
\end{document}

```

\pptToc will render the table of contents in an interactive “clickable” format. Thanks to the use of **crumbs**, there will be a navigation at the top left corner.

\pptSection Then, put sections inside chapters:

```

\begin{document}
\pptToc
\plush{\pptChapter{About Me}}
\plush{\pptSection[Student]{I'm a Student}}
...
\plush{\pptSection[Athlete]{Also, I'm an Athlete}}
...
\plush{\pptChapter[Idea]{My Idea Is Novel}}
\plush{\pptSection{Novelty}}
\plush{\pptSection{Impact}}
\end{document}

```

The \pptChapter and \pptSection commands used together will render nice two-level navigation menu at the top left corner.

\pptLeft You can place the title of the presentation and your name at the bottom of each slide, on the left and on the right respectively:

```

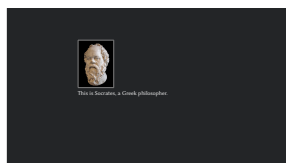
\documentclass{article}
\usepackage{clicks}
\usepackage[template,scheme=light]{ppt-slides}
\pptLeft{How Did I Spend Last Summer}
\pptRight{Yegor Bugayenko}
\begin{document}
...
\end{document}

```

If you don't use the \pptLeft or \pptRight commands, nothing will be printed at the bottom.

3 Commands

\pptPic You can add an image to the slide (the first argument is the width of the image in relation to \textwidth, while the second one is the path of it):

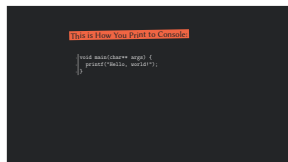


```

1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \pptPic{0.2}{socrates.jpg} \\
5 This is Socrates, a Greek philosopher.
6 \end{document}

```

`\pptHeader` You can add a piece of code to the slide (we recommend using [ffcode](#)):



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \usepackage{ffcode}
4 \begin{document}
5 \pptHeader{This is How You Print to Console:}
6 \begin{ffcode}
7 void main(char** args) {
8     printf("Hello, world!");
9 }
10 \end{ffcode}
11 \end{document}
```

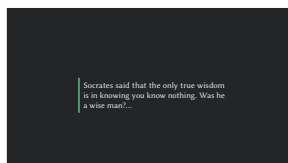
Don't use `\plick` or `\plush`, they won't work with code snippets. The `\pptHeader` command prints a header similar to what the `\pptSection` command prints, but doesn't start a new section.

`\pptQuote` You can put a quote next to the image, by using the `\pptQuote` command:
`\pptBanner`



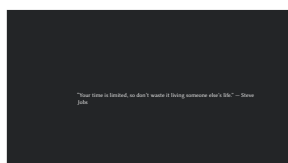
```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \pptBanner{My Favorite Quote}
5 \pptQuote{socrates.jpg}{The only true wisdom
6 is in knowing you know nothing}{Socrates}
7 \end{document}
```

`\pptThought` For an interesting thought in the middle of the slide, use `\pptThought`:



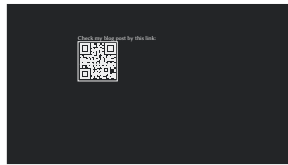
```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \pptThought{Socrates said that the only true
5 wisdom is in knowing you know nothing. Was he
6 a wise man?...}
7 \end{document}
```

`pptMiddle` You can put your content vertically in the middle of the page, using the `pptMiddle` environment:



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \begin{pptMiddle}
5 ‘‘Your time is limited, so don't waste it
6 living someone else's life.’’ --- Steve Jobs
7 \end{pptMiddle}
8 \end{document}
```

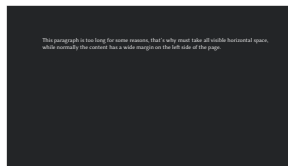
`\pptQR` Sometimes it's convenient to show a QR code to your audience instead of a URL, since it's easier to use — they can scan it:



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 Check my blog post by this link: \\
5 \pptQR{https://www.yegor256.com}
6 \end{document}
```

pptWide
pptWideOne

The code will look like this, thanks to [qrcode](#) package that we use behind the scene. Sometimes you need your slide content to take all visible horizontal space:



```
1 \documentclass{article}
2 \usepackage[template,scheme=dark]{ppt-slides}
3 \begin{document}
4 \begin{pptWideOne}
5 This paragraph is too long for some reasons,
6 that's why must take all visible horizontal
7 space, while normally the content has a wide
8 margin on the left side of the page.
9 \end{pptWideOne}
10 \end{document}
```

You can also use `\begin{pptWide}{X}`, where `X` is the number of columns to render.

4 Package Options

template You start with a template for your slide deck:

```
\documentclass{article}
\usepackage[template=9x6]{ppt-slides}
\begin{document}
...
\end{document}
```

There is only one template, which is used by default: `9x6`. If you don't specify the name, it will be used. If you don't use the `template` option at all, the default `article` will be rendered, which is not what you want.

scheme You can choose a color scheme for your slides, using the `scheme` option of the package:

```
\usepackage[template,scheme=light]{ppt-slides}
```

There are a few out-of-the-box schemes available: `light`, `dark`, `light-mono`, and `dark-mono`. You can design your own, using `ppt-light.tex` file as an example:

```
\usepackage[template,scheme=/usr/local/my-colors.tex]{ppt-slides}
```

directory You can change the location of the templates and schemes by means of the `directory` package option:

```
\usepackage[directory=/tmp,template=foo]{ppt-slides}
```

nominutes By default, if you render your slide deck in non-static mode (the `static` option for the “clicks” package), there will be minutes tracking in the right top corner of each slide. You can turn this off by using the `nominutes` option of the package:

```
\usepackage[nominutes]{ppt-slides}
```

5 Implementation

First, we parse the package options:

```
1 \RequirePackage{pgfopts}
2 \makeatletter
3 \pgfkeys{
4   /ppt-slides/.is family,
5   /ppt-slides,
6   template/.estore in = \ppt@template,
7   template/.default = 9x6,
8   scheme/.estore in = \ppt@scheme,
9   directory/.estore in = \ppt@directory,
10  directory/.default = ,
11  nomminutes/.estore in = \ppt@nomminutes,
12  directory
13 }
14 \ProcessPgfPackageOptions{/ppt-slides}
15 \makeatother
```

Then, we include the template:

```
16 \makeatletter\ifdefined\ppt@template
17   \input{\ppt@directory ppt-templates/ppt-\ppt@template.tex}
18 \else
19   \message{^^Jppt: No template is loaded, because
20     the 'template' option is not specified}
21 \fi\makeatother
```

Then, we include the scheme:

```
22 \makeatletter\ifdefined\ppt@scheme
23   \RequirePackage{xcolor}
24   \input{\ppt@directory ppt-schemes/ppt-\ppt@scheme.tex}
25 \else
26   \message{^^Jppt: No color scheme is loaded, because
27     the 'scheme' option is not specified}
28 \fi\makeatother
```

Then, we include a few packages:

```
29 \RequirePackage{ifthen} % for if/then/else in LaTeX
30 \RequirePackage{href-ul} % for nice hyperlinks
31 \RequirePackage{pagecolor} % to change page color
32 \RequirePackage{varwidth} % for pptWide
33 \RequirePackage{qrcode} % to print QR codes
34 \RequirePackage{tikz} % for drawings in TOC
35 \usetikzlibrary{calc} % for positioning of pinned boxes
36 \RequirePackage{tikzpagenodes} % for pptPin
37 \RequirePackage[inline]{enumitem} % for lists in crumbs
38 \RequirePackage{crumbs} % for crumbs
```

`\pptMiddle` Then, we define the `\pptMiddle` environment:

```
39 \newenvironment{pptMiddle}
40   {\vspace*{\fill}}
41   {\vspace*{\fill}}
```

`\pptBanner` Then, we define the `\pptBanner` command:

```
42 \newcommand\pptBanner[2][green]{%
```

```

43 \par%
44 {\setlength{\fboxsep}{6pt}%
45 \colorbox{#1}{\color{\thepagecolor}\large #2}}
46 \par%
47 }

```

`\pptChapter` Then, we define the `\pptChapter` command:

```

48 \newcommand\pptChapter[2] [] {%
49 \crumbection[#1]{#2}%
50 \begin{pptMiddle}%
51 {\large\ttfamily Chapter \#\the\value{section}:}%
52 \newline%
53 \rotatebox{1}{\pptBanner [blue]{\LARGE#2}}%
54 \end{pptMiddle}%
55 }

```

`\pptSection` Then, we define the `\pptSection` command:

```

56 \newcommand\pptSection[2] [] {%
57 \subcrumbection[#1]{#2}%
58 \pptHeader{#2}%
59 }

```

`\pptHeader` Then, we define the `\pptHeader` command:

```

60 \newcommand\pptHeader[1] {%
61 \vspace*{-0.75in}\hspace*{-0.5in}%
62 \rotatebox{1}{\pptBanner [orange]{\Large #1}}\vspace*{0.15in}%
63 \par%
64 }

```

`\pptTitle` Then, we define the `\pptTitle` command:

```

65 \newcommand\pptTitle[2] {%
66 {\pptBanner{\Huge #1}}%
67 {\pptBanner{\large #2}}%
68 }

```

`\pptToc` Then, we define the `\pptToc` command:

```

69 \newcommand*\pptTOC{}
70 \newcommand\pptToc[1] [] {%
71 \renewcommand*\pptTOC{}%
72 \print{%
73 \def\param{#1}%
74 \ifx\param\empty\else%
75 \pptBanner{#1}%
76 \fi%
77 \tikz{%
78 \node (z) {%
79 \begin{varwidth}{\textwidth}%
80 \ifx\pptTOC\empty\else%
81 \begin{itemize}[label={}]
82 \pptToc
83 \end{itemize}
84 \fi%
85 \end{varwidth}%

```

```

86     };%
87     \path [draw=blue,line width=4pt] (z.north west) -- (z.south west);%
88   }%
89 }%
90 \begin{@empty}%
91   \renewcommand\crumb[2]{%
92     \ifx\pptTOC\empty\else\ifdefined\click\click[0]\fi\fi
93     \gappto\pptTOC{\item ##2}
94   }
95   \crumbs
96 \end{@empty}%
97 \ifdefined\flush\flush[1]\fi%
98 }

```

\pptQuote Then, we define the \pptQuote command:

```

99 \RequirePackage{tabularx}
100 \newcommand\pptQuote[3]{%
101   \begin{tabularx}{\columnwidth}{c>{\raggedright\arraybackslash}X}%
102     \raisebox{\dimexpr-\height+\ht\strutbox}{\pptPic{0.25}{#1}}%
103     &%
104     ‘‘#2’’%
105   \def\param{#3}%
106   \ifx\param\empty\else%
107     \newline\newline%
108     \small%
109     --- #3%
110   \fi%
111   \\%
112 \end{tabularx}%
113 }

```

\pptPic Then, we define the \pptPic command:

```

114 \newcommand\pptPic[2]{%
115   {%
116     \setlength{\fboxsep}{0pt}%
117     \setlength{\fboxrule}{1pt}%
118     \fcolorbox{gray}{white}{%
119       \includegraphics[width=#1\columnwidth]{#2}%
120     }%
121   }%
122 }

```

\pptPin Then, we define the \pptPin command:

```

123 \newcommand\pptPin[2][right]{%
124   \begin{tikzpicture}[remember picture,overlay]%
125     \ifthenelse{\equal{#1}{left}}{%
126       {%
127         \node[anchor=north east] at
128           ($(\current page text area.north west) - (24pt, 18pt)$) {%
129           \begin{minipage}{0.25\textwidth}%
130             \raggedleft #2%
131           \end{minipage}%
132         };%
133       }%

```

```

134   {%
135     \node[anchor=north east] at (current page text area.north east) {%
136       \begin{minipage}{0.25\textwidth}%
137         \raggedright #2%
138       \end{minipage}%
139     };%
140   }%
141 \end{tikzpicture}%
142 }

```

`\pptThought` Then, we define the `\pptThought` command:

```

143 \newcommand\pptThought[1]{%
144   \begin{pptMiddle}%
145     \tikz{%
146       \node [inner sep=18pt] (z) {%
147         \begin{varwidth}{0.8\textwidth}
148           \Large
149             #1
150         \end{varwidth}
151       };
152       \path [draw=green,line width=8pt] (z.north west) -- (z.south west);
153     }%
154   \end{pptMiddle}%
155 }

```

`\pptSnippet` Then, we define the `\pptSnippet` command:

```

156 \newcommand\pptSnippet[2][\small]{%
157   \begin{samepage}%
158     #1\verbatiminput{#2}%
159   \end{samepage}%
160 }

```

`\pptQR` Then, we define the `\pptQR` command:

```

161 \newcommand\pptQR[2][2in]{%
162   \tikz{%
163     \node[draw=white]{%
164       \href{#2}{%
165         \qrcode[height=#1]{#2}%
166       }%
167     }%
168   }%
169 }

```

`\pptPinQR` Then, we define the `\pptPinQR` command:

```

170 \RequirePackage{seqsplit}
171 \newcommand\pptPinQR[2][right]{%
172   \pptPin[#1]{%
173     \begin{minipage}{1.4in}%
174       \raggedleft%
175       \pptQR[1in]{#2}%
176       \\[3pt]
177       \ttfamily\fontsize{8pt}{10pt}\selectfont%
178       \seqsplit{#2} \href{#2}{\rightarrow}%
179     }

```



```

180 \end{minipage}%
181 }%
182 }

```

pptWide Finally, we define the pptWide and pptWideOne environments:

```

183 \newenvironment{pptWideOne}
184 {\begin{adjustwidth}{-2in}{-1in}}
185 {\end{adjustwidth}}
186 \newenvironment{pptWide}[1]
187 {\begin{pptWideOne}\begin{multicols}{#1}}
188 {\end{multicols}\end{pptWideOne}}

```

Change History

0.0.1	General: First draft. 5	0.2.0	General: We migrated to the ltxdoc format and .dtx file. 5
0.1.4	General: The nominutes package option added. 5	0.2.1	\pptMiddle: The \pptMiddle command turned into pptMiddle environment. 5
0.1.5	General: The default directory fixed. . . 5		

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols	\dimexpr 102	\href 164, 178
\# 51	E	\hspace 61
A	\empty 74, 80, 92, 106	\ht 102
\arraybackslash . . . 101	\end 54, 83, 85, 96, 112, 131, 138, 141, 150, 154, 159, 180, 185, 188	\Huge 66
B	\equal 125	I
\begin 50, 79, 81, 90, 101, 124, 129, 136, 144, 147, 157, 173, 184, 187	F	\ifdefined . . 16, 22, 92, 97
C	\fboxrule 117	\ifthenelse 125
\click 92	\fboxsep 44, 116	\ifx 74, 80, 92, 106
\color 45	\fcolorbox 118	\includegraphics . . 119
\colorbox 45	\fill 40, 41	\input 17, 24
\columnwidth . . . 101, 119	\flush 97	\item 93
\crumb 91	\fontsize 177	L
\crumbaction 49	G	\LARGE 53
\crumbs 95	\gappto 93	\Large 62, 148
D	H	\large 45, 51, 67
\def 73, 105	\height 102	M
		\makeatletter . . . 2, 16, 22
		\makeatother . . . 15, 21, 28
		\message 19, 26

N			
<code>\newcommand</code>		<code>\rightarrow</code>	
... 42, 48, 56, 60,	<code>\pptPin</code>	178	
65, 69, 70, 100, 114,	<code>\pptPinQR</code>	<code>\rotatebox</code>	
123, 143, 156, 161, 171	<code>\pptQR</code>	53, 62	
<code>\newenvironment</code> ...	<code>\pptQuote</code>	S	
..... 39, 183, 186	<code>\pptSection</code>	<code>\selectfont</code>	177
<code>\newline</code>	<code>\pptSnippet</code>	<code>\seqsplit</code>	178
52, 107	<code>\pptThought</code>	<code>\setlength</code> ..	44, 116, 117
<code>\node</code> . 78, 127, 135, 146, 163	<code>\pptTitle</code>	<code>\small</code>	108, 156
	<code>\pptTOC</code> 69, 71, 80, 82, 92, 93	<code>\strutbox</code>	102
	<code>\pptToc</code>	<code>\subcrumbsection</code>	57
	<code>\pptWide</code>	T	
P		<code>\textwidth</code> 79, 129, 136, 147	
<code>\par</code>	<code>\print</code>	<code>\the</code>	51
43, 46, 63	72	<code>\thepagecolor</code>	45
<code>\param</code> 73, 74, 105, 106	<code>\ProcessPgfPackageOptions</code>	<code>\tikz</code>	77, 145, 162
<code>\path</code> 14	<code>\ttfamily</code>	51, 177
87, 152		U	
<code>\pgfkeys</code>	Q		<code>\usetikzlibrary</code>
3			35
<code>\ppt@directory</code> . 9, 17, 24		V	
<code>\ppt@nominutes</code>	<code>\qrcode</code>	<code>\value</code>	51
11	165	<code>\verbatiminput</code>	158
<code>\ppt@scheme</code> 8, 22, 24	R		<code>\vspace</code> 40, 41, 61, 62
<code>\ppt@template</code> .. 6, 16, 17	<code>\raggedleft</code>		
<code>\pptBanner</code>	130, 174		
. 42, 53, 62, 66, 67, 75	<code>\raggedright</code> ...		
<code>\pptChapter</code>	101, 137		
48	<code>\raisebox</code>		
<code>\pptHeader</code>	102		
58, 60	<code>\renewcommand</code>		
<code>\pptMiddle</code>	71, 91		
39	<code>\RequirePackage</code> 1, 23,		
<code>\pptPic</code>	29, 30, 31, 32, 33,		
102, 114	34, 36, 37, 38, 99, 170		